Optimasi Training pada Jaringan Syaraf Tiruan menggunakan Algoritma Extended Kalman Filter

Zaqiatud Darojah
zaqiah@eepis-its.edu

Abstrak

Proses training pada jaringan syaraf tiruan (JST) feedforward menggunakan algoritma konvensiona, seperti algoritma backpropagation mempunyai kelemahan dalam mencari bobot-bobot yang konvergen, yaitu kekonvergenan berjalan dengan lambat dan dilakukan cara coba-coba dalam menentukan parameter-parameter yang sesuai. Penentuan parameter-parameter yang sesuai dapat menghambat pekerjaan dan tidak praktis dalam memperoleh hasil yang cepat. Permasalahan tersebut dapat diselesaikan menggunakan teknik optimasi nonlinear sehingga didapatkan bobot-bobot yang konvergen dengan cepat (fast learning). Salah satunya adalah metode extended Kalman filter. Penerapan algoritma extended Kalman filter sebagai metode training JST dapat dilakukan dengan memformulasikan JST sebagai konsep state space yang mirip dengan sistem dinamik nonlinear. Hasil penelitian menunjukkan bahwa algoritma EKF memiliki kecepatan konvergensi yang lebih cepat dibandingkan dengan algoritma pembanding, dalam hal ini adalah algoritma bavkpropagation.

Kata Kunci:

1. Pendahuluan

Jaringan syaraf tiruan (JST) adalah suatu sistem pemroses informasi yang mempunyai karakteristik seperti halnya jaringan syaraf biologi. Karakteristik yang menarik dari JST ini adalah kemampunnya untuk belajar (learning/training). Proses training pada JST betujuan untuk mencari bobot-bobot yang konvergen antar lapisan sedemikian hingga bobot-bobot yang diperoleh menghasilkan output yang diinginkan [4].

Permasalahan training pada JST dalam upaya membentuk jaringan yang mampu mengenali pola data yang diberikan merupakan permasalahan tersendiri bagi peneliti yang akan menerapakan JST dalam suatu teknologi. Hal ini disebabkan belum adanya metoda baku yang bisa dijadikan pedoman oleh peneliti dalam menerapkan struktur jaringan terbaik yang mampu megenali pola dengan akurat tanpa melakukan *trial and error* dalam memperoleh jaringan yang optimal.

Jaringan syaraf tiruan *feedforward* merupakan salah satu arsitektur dari JST yang banyak digunakan dalam aplikasi di berbagai bidang. Proses training jaringan *feedforward* menggunakan algoritma *backpropagataion* konvensional mempunyai kelemahan dalam hal mencari bobot yang konvergen. Pada algoritma tersebut kekonvergenan berjalan dengan lambat dan membutuhkan parameter yang sesuai dengan cara coba-coba (*guesswork*). Banyak teknik yang disarankan untuk mengembangkan kecepatan konvergensi pada proses training dari algoritma backpropagation, seperti teknik optimasi gradient descent [6]. Pengembangan - pengembangan yang dilakukan secara komputasi tidak mahal dan biasanya mencapai optimum lokal, akan tetapi teknik-teknik tersebut membutuhkan parameter yang sesuai, sehingga digunakan cara coba-coba dalam menentukan parameter yang sesuai [2, 5]..

Permasalahan komputasi dalam mencari bobot pada jaringan *feedforward* untuk menyelesaikan pemetaan input atau output yang diinginkan dari R^N ke $(0,1)^L$ dapat digambarkan sebagai permasalahan sistem identifikasi nonlinear berdimensi tinggi. Permasalahan tersebut dapat diselesaikan menggunakan teknik optimasi nonlinear [5]. Pada tahun 1989 Singhal dan Wu menyarankan penggunaan algoritma extended Kalman filter untuk *fast learning*, yaitu proses training berjalan dengan cepat dalam memperoleh konvergensi bobot. Dengan algoritma extended Kalman filter ini, proses training pada jaringan *feedforward* dapat dilihat sebagai estimasi state sistem dinamik nonlinear [2].

Penggunaan algoritma extended Kalman filter menunjukkan sebagai metode training JST memiliki banyak keuntungan. Penelitian Lary et al, 2004, menunjukkan bahwa hanya dibutuhkan 3 epoch (iterasi) untuk proses training menggunakan EKF dibandingkan dengan penelitian sebelumnya pada data yang sama yang membutuhkan 1 juta iterasi. Hasil penelitian tersebut menunjukkan bahwa algoritma EKF sebagai metode training JST memiliki performansi konvergensi yang sangat cepat.

Penelitian ini akan membahas kembali penerapan EKF sebagai metoda training JST *feedforward* yang diaplikasian pada peramalan data runtun waktu (*time series*).

2. Algoritma Extended Kalman Filter

Kalman Filter merupakan metode estimasi variable keadaan (*state*) dari suatu sistem dinamik dengan melakukan prediksi *state* tersebut dan selanjutnya dilakukan pengukuran untuk mengoreksi state secara reekursif dengan cara meminimumkan kovarian eror penaksiran. Kalman Filter memberikan solusi yang baik dalam mengestimasi variable keadaan untuk sistem dinamik linear [3]. Sedangkan jika sistem dinamiknya berupa nonlinear, maka dilakukan perluasan terhadap algoritma Kalman filter melalui prosedur linierisasi pada tiap waktu sehingga didapatkan suatu algoritma yang dikenal dengan algoritma extended Kalman filter.

Pandang sistem dinamik stokastik nonlinear berikut:

$$x_{k+1} = f(x_k, u_k) + w_k \quad (1)$$

$$z_k = h(x_k) + v_k \quad (2)$$

dimana $x_k \in R^n$ adalah variabel keadaan (*state*), $z_k \in R^m$ adalah variabel pengukuran dengan asumsi $w_k \sim N(0,Q)$, $v_k \sim N(0,R)$ dan $Q > 0$, $R > 0$, {w(k)} dan {v(k)} tidak berkorelasi satu sama lain. $f(x_k, u_k)$ melambangkan matriks transisi nonlinear, h(x_k) melambangkan matriks pengukuran nonlinear.

Ide dasar dari extended Kalman filter adalah melinearisasi model pada Persamaan (1) dan Persamaan (2) pada tiap waktu disekitar estimasi state $\hat{x}_k$. Setiap kali model linear didapatkan, persamaan Kalman filter standar diterapkan [1].

Sama halnya dengan algoritma Kalman filter standar, formulasi dari algoritma extended Kalman filter diekspresikan dalam dua tahap, yaitu tahap *time update* (*forecast step*) dan tahap *measurement update* (*analysis step*). Tahap *time update* merupakan tahap dimana informasi plant pada sistem digunakan. Pada tahap ini menghasilkan nilai prediksi *state* dan nilai prediksi kovarian eror. Tahap *measurement update* merupakan tahap dimana informasi pengukuran digunakan. Pada tahap ini menghasilkan nilai koreksi berdasarkan data pengukuran terbaru [3]. Kedua tahapan ini diperoleh dengan menurunkan perambatan mean dan kovarian variabel state pada Persamaan (1) dan variabel pengukuran pada Persamaan (2). Algoritma extended Kalman filter secara lengkap ditulis sebagai berikut:

Inisialisasi:

$$\hat{x}_0^a = \bar{x}_0, \qquad P_0^a = P_{x0}$$

Time Update:

Estimasi : $\hat{x}_{k+1}^f = f(k, \hat{x}_k^a)$

Kovarian eror : $P_{k+1}^f = A_k P_k^a A_k^T + Q$

Measurement Update:

Estimasi : $\hat{x}_k^a = \hat{x}_k^f + K_k[z_k - h(\hat{x}_k^f)]$

Kalman Gain : $K_K = P_k^f H_k^T [H_k p_{K+1}^f H^T + R]^{-1}$

Kovarian eror : $P_k^a = [I - K_k H_k] P_k^f$

Jacobian:

$$A_k = \left.\frac{\partial f(x,u)}{\partial x}\right|_{x = \hat{x}_k^a} \text{ dan } H_k = \left.\frac{\partial h(x)}{\partial x}\right|_{\hat{x} = x_k^f}$$

Notasi $\hat{x}_k^f$ menunjukkan nilai estimasi state ke-k pada tahap *time update* (*forecast step*), sedangkan notasi $\hat{x}_k^a$ menunjukkan nilai estimasi state ke-k pada tahap *time update* (*analysis step*).

3. Algoritma Extended Kalman Filter Sebagai Metode Training JST

Singhal and Wu (1989) adalah orang pertama yang menyarankan penggunaan EKF untuk teknik training pada JST *feedforward*. Argumen ini didasarkan pada alasan-alasan sebagai berikut (Lary and Musa, 2004):

- Jaringan syaraf multi*layer feedforward* dapat digambarkan sebagai suatu sistem dinamik nonlinear yang *state*nya berupa suatu vektor bobot.
- *Training* dari JST dapat dipandang sebagai permasalahan estimasi *state* untuk sistem nonlinear.
- Kalman filter dikenal sebagai metode yang memberikan estimasi optimal dari *state* sistem dinamik linear. Pada perkembangannya juga dikenal algoritma Kalman filter versi extended yang dapat digunakan untuk mengestimasi *state* sistem dinamik nonlinear.
- JIka JST diformulasikan sebagai konsep *state* space yang mirip dengan sistem dinamik nonlinear, maka dapat ditemukan vektor bobot terbaik dengan menerapkan algoritma extended Kalman filter.

Bentuk sistem dinamik dari jaringan syaraf tiruan secara metematis dapat digambarkan dengan dua persamaan berikut [2] :

$$w_{j+1} = w_j + e_j \quad (3)$$

$$d_j = h[w_j, x_j] + v_j \quad (4)$$

dengan:

j : indeks iterasi

w_j : *state* dari jaringan syaraf *feedforward*

e_j : vektor noise proses

d_j : vektor output (target)

v_j : vektor pengukuran noise
x_j : vektor input
$h[w_j, x_j]$: fungsi iteratif yang menggambarkan jaringan, nilai dari fungsi ini output dari JST

dengan asumsi bahwa v_j adalah *white noise* dengan $E[v_j v_j^T] = R_j$ matriks kovarian, e_j adalah *white noise* dengan $E[e_j e_j^T] = Q_j$ matriks kovarian dan v_j, e_j tidak saling berkorelasi untuk semua i, j.

Persamaan (1) dikenal sebagai persamaan proses yang merupakan *state* dari sistem JST dengan elemen-elemennya berupa semua koneksi bobot yang ada pada arsitektur JST. Sedangkan persamaan (2) dikenal sebagai persamaan observasi yang merupakan pengukuran dari sistem JST dengan elemennya berupa vektor target dari JST [2].

Formulasi EKF didapatkan dengan cara menurunkan mean dan kovarian dari variabel state dan pengukuran. Berikut ini adalah hasil penuruanan algoritma EKF pada sistem dinamik JST yang terdapat pada Persamaan (3) dan Persamaan (4).

Time Update:

Estimasi : $\hat{w}_j^f = \hat{w}_{j-1}^a$

Kovarian error : $P_j^f = P_{j-1}^a + Q_j$

Measurement Update:

estimasi : $\hat{w}_j^a = \hat{w}_j^f + K_j[d_j - h(\hat{w}_j^f, x_j)]$

kovarian error : $P_j^a = [I - K_j H_j]P_j^f$

Kalman gain : $K_j = P_j^f H_j^T [H_j P_j^f H_j^T + R_j]^{-1}$

dengan

$$H_j = \frac{\partial h[\hat{w}_{j-1}, x_j]}{\partial \hat{w}_{j-1}}$$

Pada kasus JST *feedforward*, kovarian noise pada state, Q_j, bernilai nol (Haykin, 2001; Lary and Musa, 2004), Sehingga algoritma EKF pada sistem dinamik JST diatas dapat diringkas menjadi satu tahap, yaitu tahap *measurement update*.

Nilai output dari JST, $h(w, x)$, merupakan variabel pengukuran dari sistem dinamik pada JST. Nilai tersebut menggambarkan perhitungan bobot-bobot pada lapisan-lapisan dalam JST. Adapun nilai output JST adalah sebagai berikut [2]:

$$\hat{y}_j = h(w, x) = \tanh[w(m(n+1)+1 : w(m(n+1)+m+1]^T z(1 : m+1)] \quad (5)$$

dimana $z(2 : m+1) = \tanh[w(1 : n+1]^T x(1 : 1 : n+1)]$, dengan x(1) dan z(1) adalah nilai bias pada masing-masing input layer dan hidden layer.

Fungsi tanh(.) merupakan fungsi aktifasi yang digunakan, dimana fungsi ini memiliki nilai yang sama dengan fungsi bipolar sigmoid (Fausset).

Pada penelitian ini, nilai dari R dihitung menggunakan faktor *forgetting* yang diperkenankan oleh Zhang and Li [2]. Adapun rumusan dari factor *forgetting* adalah:

$$\lambda_j = \lambda^0 \lambda_{j-1} + (1 - \lambda^0) \quad (6)$$

dengan λ_0 dan λ^0 adalah *tunable parameters*.

4. Rancangan Proses *Training* JST dengan Algoritma Extended Kalman Filter

Proses *training* JST menggunakan algoritma EKF bertujuan untuk mendapatkan estimasi vektor bobot yang meminimumkan kovarian eror secara reekursif. Vektor bobot yang dimaksud adalah nilai bobot dan bias yang mengkoneksikan tiap unit antar layer pada arsitektur JST.

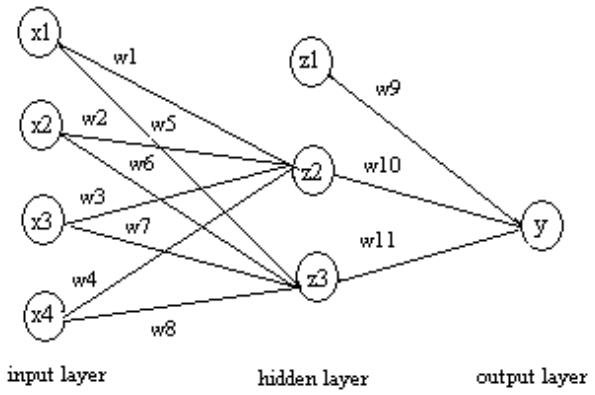
Misalkan dipunyai jaringan syaraf *feedforward* dengan arsitektur sebagai berikut:

- Satu input layer yang terdiri dari *n* unit dan satu bias
- Satu hidden layer yang terdiri dari *m* unit dan satu bias
- Satu unit output pada output layer

Pada arsitektur tersebut, dapat diperoleh informasi yang digunakan untuk prosedur *training* JST menggunakan algoritma EKF sebagai berikut:

- *w* adalah vektor bobot berukuran $[m(n+1)+m+1]\times 1$
- *x* adalah vektor input berukuran $[n+1]\times 1$, dengan x(1) berupa konstanta, yang menunjukkan bias pada input.
- *P* dan *Q* adalah vektor kovarian eror dan vektor kovarian noise sistem berukuran $[m(n+1)+m+1]\times[m(n+1)+m+1]$
- *R* dan d_j adalah vektor kovarian noise pengukuran dan vektor target berukuran 1 x 1
- *K* berukuran $[m(n+1)+m+1]\times 1$
- *H* berukuran $1\times[m(n+1)+m+1]$

Sebagai contoh, misalkan dipunyai arsitektur JST *feedforward* dengan tiga node pada input layer dan dua node pada hidden layer seperti yang ditunjukkan pada Gambar 1.



Gambar 1: Arsitektur JST *feedforward* (3, 2,1)

Dari arsitektur tersebut, akan diperoleh informasi sebagai berikut:

- w vektor boobt berukuran 13 x 1, yaitu $w = [w_1\ w_2\ w_3\ w_4 \cdots w_{10}\ w_{11}]^T$
- x berukuran 4 x 1, $x = [x_1\ x_2\ x_3 x_4]^T$, dengan x_1 berupa bias pada input layer
- P dan Q berukuran 11 x 11
- R dan d_j berukuran 1 x 1
- K berukuran 11 x 1
- H berukuran 1 x 11

Sedangakan nilai dari output JST adalah:

$$\hat{y}_j = \tanh\left([w_9\ w_{10}\ w_{11}]^T [z_1\ z_2\ z_3]\right)$$

$$= \tanh\left(\sum_{i=1}^{3} w_{i+8} z_i\right)$$

dimana nilai $z_1 = 1$, karena merupakan bias pada hidden layer. Sedangkan nilai z_2 dan z_3 adalah:

$$z_2 = \tanh\left([w_1\ w_2\ w_3\ w_4]^T [x_1\ x_2\ x_3\ x_4]\right)$$

$$= \tanh\left(\sum_{i=1}^{4} w_i z_i\right)$$

dan

$$z_3 = \tanh\left([w_5\ w_6\ w_7\ w_8]^T [x_1\ x_2\ x_3\ x_4]\right)$$

$$= \tanh\left(\sum_{i=1}^{4} w_{i+4} x_i\right)$$

Nilai x_1 berupa konstanta karena merupakan bias pada input layer (umumnya bernilai 1).

5. Simulasi dan Diskusi

Simulasi dilakukan untuk mendapatkan performa dari algoritma EKF dan juga algoritma *backpropagation* sebagai pembanding. Training JST *feedforward* pada penelitian diaplikasikan untuk prediksi data time series. Data yang digunakan adalah bilangan sunspot. Arsitektur JST yang digunakan pada setiap simulasi adalah satu unit pada input layer dan satu unit pada hidden layer.

Pada percobaan pertama, bobot-bobot pada JST diinisialisasi secara random.Inisialisasi dari parameter-parameter lain yang dibutuhkan ditetapkan dengan beberapa percobaan. Untuk melihat akurasi dari JST, digunakan nilai MSE dan nilai korelasi antara data actual dan data hasil prediksi sebagai pembanding. Tabel 1 menunjukkan hasil simulasi menggunakan lima kali iterasi dan10 kali iterasi dari masing-masing algoritma training.

Tabel 1
Hasil Simulasi dengan lima dan 10 iterasi

Algoritma	Performansi dengan 5 iterasi		Performansi dengan 10 iterasi	
	MSE	Korelasi	MSE	Korelasi
Back propagation	110.058	0.9689	16.3083	0.9972
	120.497	0.9656	4.0271	0.9992
	124.843	0.9658	111.352	0.9685
	210.086	0.9668	31.5426	0.9933
	112.138	0.9680	15.7652	0.9973
EKF	1.0164	0.9997	0.9057	0.9997
	0.5648	0.9998	0.8811	0.9998
	1.5869	0.9997	1.0150	0.9997
	3.1713	0.9992	1.4284	0.9996
	0.6431	0.9998	0.7127	0.9998

Hasil simulasi dengan 5 kali iterasi yang terdapat pada Tabel 1 menunjukkan bahwa EKF memiliki performa yang baik dibandingkan backpropagation, hal ini dapat dilihat dari nilai MSE yang dihasilkan. EKF pada simulasi 10 kali iterasi ini juga memiliki performa yang bagus dengan korelasi berkisar antara 0.9996-0.9998. Sedangkan untuk *backpropagation* nilai MSE yang dihasilkan masih sangat tinggi diabndingkan dengan EKF.

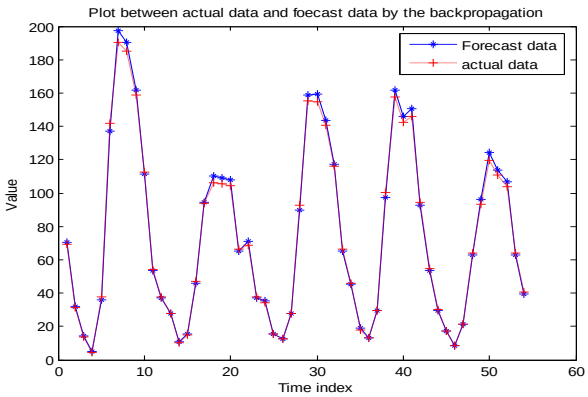
Simulasi selanjutnya adalah dengan menambahkan jumlah iterasi untuk mengetahui apakan sistem telah mencapai global minima atau local minima. Hasil simulasi dengan menambahakan 100 kali iterasi dari dua metoda training yang diterapkan adalah sebagai berikut:

Tabel 2
Hasil simulai dengan 100 kali iterasi

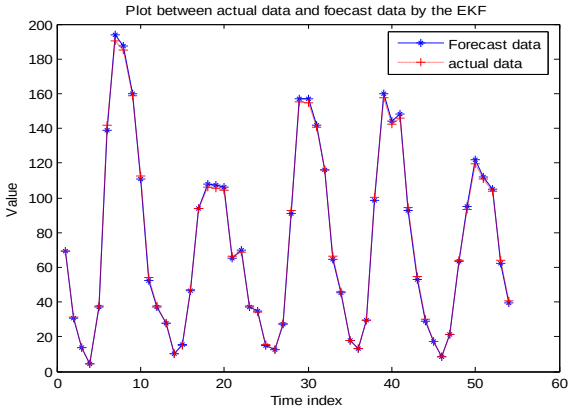
Algoritma	Performansi	
	MSE	Korelasi
Back propagation	2.4684	0.9995
	2.4547	0.9996
	2.3507	0.9996
	2.4615	0.9996
	2.4627	0.9996
EKF	0.7751	0.9998
	0.6192	0.9998
	0.8153	0.9998
	0.7775	0.9998
	2.4684	0.9995

Tabel 2 menunjukkan bahwa algoritma *backpropagation* mengalami perbaikan performa yang lebih baik dan lebih konsisten. Akan tetapi nilai MSE dan korelasi antara data actual dan data hasil prediksi JST masih jauh fdari algoritma EKF. Hasil simulasi EKF pada kasus ini tidak terlalu jauh berbeda dengan simulasi yang menggunakan 10 kali iterasi. Dengan kata lain, algoritma EKF mempunyai konvergensi lebih cepat dibandingkan dengan algoritma backpropagation.

Untuk menggambarkan hasil simulasi diata, berikut ini akan ditampilkan beberapa plot grafik antara data aktual dengan data hasil prediksi dari JST untuk masing-masing algoritma training yang digunakan.



Gambar 2 : Plot Data Aktual dengan Data Prediksi Menggunakan *Backpropagation*



Gambar 2 : Plot Data Aktual dengan Data Prediksi Menggunakan EKF

6. Kesimpulan

Pada penelitian ini telah dibahas kerangka umum training JST menggunakan teori optimal filtering, yaitu algoritma EKF. Konsep penerapan algoritma EKFsebagai metoda training JST *feedforward* dalam penelitian ini didasarkan pada pendekatan yang digunakan oleh Singhal and Wu dalam menerapkan EKF sebagai metoda training JST. Bobot-bobot yang ada pada jaringan feedforwarrd dipandang sebagai state dari sistem dinamik yang diestimasi secara reekursif. Untuk mengetahui performansi dari algoritma training yang digunakan, pada penelitian ini dilakukan perbandingan antara algoritma EKF dan algoritma *backpropagation*. Berdasakan hasil simulasi, EKF memiliki tingkat konvergensi lebih cepat daripada algoritma *backpropagation*.

Referensi

[1] Haykin, S. "Kalman Filters*"*,in *Kalman Filtering and Neural Networks*, ed. Haykin, S., John Wiley & Sons, USA, hal. 16-19, 2001.

[2] Lary, D.J, Mussa H.Y, "Using an Extended Kalman Filter Learning Algorithm for Feedforward Neural Network to Describe Tracer Correlations*"*, *Atmospheric Chemistry and Physics Discussion*, Vol. 4, hal. 3653-3667, 2004.

[3] Lewis, F.L, *Optimal Estimation*, John Wiley & Sons, Inc., New York, 1986

[4] Fausset, L., *Fundamental of Neural Network: Theory and Application*, Prentice Hall, Singapure, 1996.

[5] Shah, S., Palmieri, F., Datum, M., "Optimal Filtering Algorithms for Fast Learning in Feedforward Neural Network*"*, *Neural Networks Research*, Vol. 5, hal. 779-787, 1992.

[6] Kusumadewi, Sri, *Membangun Jaringan Syaraf Tiruan menggunakan Matlab & Excel Link*, Graha Ilmu, Yogyakarta, 2004.1